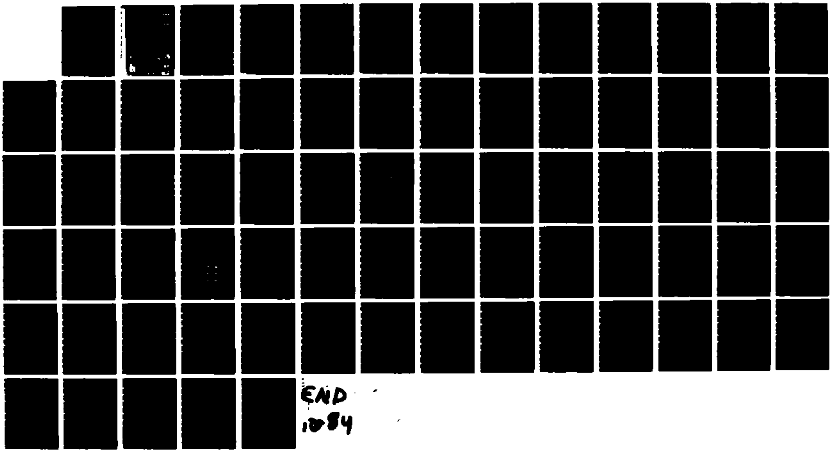
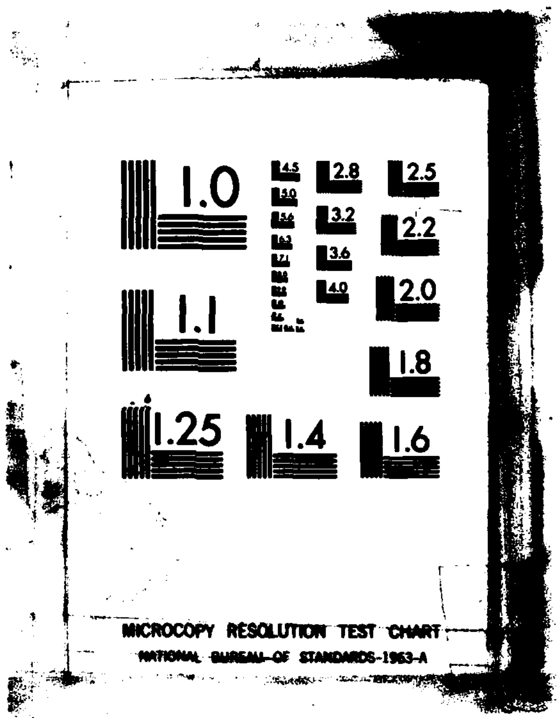
AD-A140 862 A MODEL OF HUMAN PROBLEM SOLVING IN DYNAMIC ENVIRONMENTS(U) GEORGIA INST OF TECH ATLANTA SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING A E KNAEUPER AUG 83 83-3 1/1

UNCLASSIFIED F/G 5/8 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

center for man-machine systems research

AD-A140 862

a model of human problem solving in dynamic environments

annette e. knaeuper

report no. 83-3

august 1983

DTIC
ELECTE
MAY 8
S
A

This docu...
for publi...
distribu...

school of industrial and systems engineering
georgia institute of technology
a unit of the university system of georgia
atlanta, georgia 30332

A RULE-BASED MODEL OF HUMAN PROBLEM SOLVING BEHAVIOR IN DYNAMIC ENVIRONMENTS

A THESIS

Presented to

The Faculty of the Division of Graduate Studies

By

Annette Knaeuper

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Industrial Engineering

Georgia Institute of Technology

August, 1983

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A1

A Rule-Based Model of Human Problem Solving Behavior
in Dynamic Environments

Annette Knaeuper

65 Pages

Directed by Dr. William B. Rouse

The increased use of automation in aircraft, ships, process plants, transportation networks, and other large-scale systems is changing the human's role in such systems. The manual activities of the human operator are increasingly supplanted by monitoring of automation and occasional problem solving activities. This thesis focuses on these problem solving activities.

Models of human problem solving are reviewed, with emphasis on those applicable to situations involving human-machine interaction in detecting, diagnosing, and compensation for system failures. From this review it emerges that most models developed thus far focus on a single aspect of problem solving. An overall model is presented, which considers the full breadth and robustness of human problem solving behavior in dynamic environments. A realization of the general structure of this model within a particular rule-based computer program is discussed. Results are presented from applying this program to modeling human problem solving in a process control task.

William B. Rouse

William B. Rouse, Advisor

ACKNOWLEDGEMENTS

I wish to thank my advisor, William B. Rouse, for his encouragement, support and excellent guidance throughout this research. I also like to thank my friends and colleagues at the Center for Man-Machine Systems Research: T. Govindaraj, John Hammer, Ed Brown, Deborah Eubanks, Dick Henneman, Greg Hopkins, Jody Johnson, Teresa Mann, Nancy Morris, Russell Mullenix, David Su, Eduardo Viteri and Wan Yoon. I extend a special thanks to my friend and officemate Dave Morehead. He, in particular, and my other colleagues helped and encouraged me to adjust and enjoy life and graduate study in this foreign country. The opportunity to be associated with all the aforementioned people was the most important aspect of my graduate study and stay in the United States.

A special thanks goes to Prof. Rainer Bernotat, my director at the Forschungsinstitut fuer Anthropotechnik (FAT), F.R. of Germany, who supported my wish to attend Georgia Tech and made it possible by granting me a leave of absence from the Institute. Furthermore, I would like to acknowledge my friends and colleagues, at the FAT, who encouraged me to attend graduate school.

Finally, I wish to thank deeply my parents, Agnes and KARL Knaeuper ("the human problem solver"), and my six sisters and brothers for their unwavering support of my work and encouragement while I was staying so far away from home. The love of my family means the most to me.

This research was supported by the Office of Naval Research under Work Unit NR 154-491 (Contract N00014-82-K-0487). Dr. Henry M. Halff is Contract Monitor.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF ILLUSTRATIONS

CHAPTER I

INTRODUCTION

Due to increasing automation in aircraft, ships, process plants, transportation networks, and other large-scale systems, the human's manual activities are more and more supplanted by monitoring of automation and occasional problem solving activities. This thesis focuses on these problem solving activities.

Problem solving can be characterized as a mixture of planning and control (Hayes-Roth and Hayes-Roth, 1979). Planning involves searching for a sequence of actions that will potentially lead to problem solution. Control includes execution and monitoring of this sequence of actions.

Over the last fifteen years a wide range of models of human problem solving behavior has been developed. Some of these models are reviewed in Chapter II. Most of these models focus on a single aspect of problem solving. Thus, there is a need for a general model of human problem solving, especially for dealing with dynamic systems. It is the purpose of this thesis to present such a model.

In the second section of Chapter II the nature of human expertise and some ideas of how it should be represented are considered. This leads to a discussion of rule-based modeling which seems especially well suited to modeling the human problem solver in dynamic environments.

Chapter III describes the general tasks of an operator when controlling a dynamic process as well as levels of problem solving behavior in such tasks. This leads to an outline of an overall model of human problem solving in dynamic environments, which has been proposed by Rouse (1982).

Chapter IV discusses a realization of this general structure within a particular rule-based computer program. This model builds on previously developed models and is general in that it is applicable in different problem solving environments by exchanging its problem-specific knowledge base.

Chapter V represents the application of this model in a process control simulation. Extensive data from an experiment performed by Morris (1983) were used to develop the rules for the model and to evaluate the resulting model behavior and performance. Finally, in Chapter VI, conclusions are drawn and suggestions for future research are given.

CHAPTER II

THE LITERATURE

The literature relevant for this work comes from two main areas of research: 1) modeling of human problem solving and 2) representing human knowledge. The first section of this chapter reviews several models of human problem solving which are applicable to human-machine interaction in the control of various processes. Particular consideration is given to those models which deal with detection, diagnosis and compensation for system failures and which take into account dynamic aspects of problem solving tasks. The second section of this chapter discusses representation of human knowledge and characteristics of rule-based models, which form the framework of the general model of human problem solving proposed in this thesis.

Models of Human Problem Solving

During the past ten to fifteen years a wide range of models of human problem solving behavior has been developed. Some models focus on the pattern recognition nature of human behavior in problem solving tasks. For example, familiar scripts (Schank and Abelson, 1977) or frames (Minsky, 1975) may evoke a sense of having seen a particular type of problem before. Other models use a strategic approach, e.g. symptomatic versus topographic strategies (Rasmussen, 1978).

Failure detection is typically modeled by assuming that the operator compares the actual system state with a mental model of what the process should be. Several authors have modeled human behavior in failure detection tasks. e.g., by means of signal detection theory (Sheridan and Ferrell, 1974), thresholds for error and error-rate (Niemela and Krendel, 1975) or pattern recognition methods (Greenstein and Rouse, 1982, Curry, 1981). Models based on pattern recognition assume the human's detection task to be recognizing when the pattern of features is abnormal. This type of model seems quite appropriate for real life tasks since it does not require as much explicit information as the other models.

Human performance in failure diagnosis has also been considered by several researchers. Rouse, et al. (1980) developed a rule-based model that predicted the sequence of actions chosen by the troubleshooter in a failure diagnosis task. By appropriate choices of rules and rank ordering, they were able to obtain a high level of agreement between the behavior of the model and that of the humans.

Rasmussen (1981) suggested a dichotomy of problem solving in terms of search strategies. Using topographic search, the human operator, while tracing the process through the system, may compare what is observed to a template for "normal" operation and note discrepancies between the two. With the symptomatic search, the operator may compare current system state to a number of mentally stored templates and look for a match.

Hunt and Rouse (1982) incorporated Rasmussen's concepts of symptomatic and topographic strategies into a fuzzy rule-based model of fault diagnosis. In their formulation, the type of heuristic used at a particular time was viewed as being related to the type of strategy being implemented, with rules used in a symptomatic strategy being more context specific. This model was reasonably successful in predicting the sequence of actions chosen by mechanics in troubleshooting simulated powerplants and avionics systems.

If a failure must be diagnosed during system operation, as opposed to during maintenance, then the human problem solver typically must be concerned with both keeping the system operating and diagnosing the source of the problem. Compensation and diagnosis can be viewed as two separate tasks competing for the operator's attention (Rouse and Morris, 1981a, 1981b). Unfortunately, this situation can result in the operator focusing on one task to the exclusion of the other. This may have disastrous consequences. Compensation and diagnosis are interdependent tasks, i.e., possibly conflicting or complementary, which increases the potential complexity of dealing with problem solving at multiple levels.

For those situations where compensation involves executing standard procedures or manual control, there are a variety of models available (Sheridan and Ferrell, 1974, Rouse, 1980, 1981). However, these are not models of human problem solving.

Rasmussen (1980) developed a model, where he includes the stages in the decision making process that are typical for most analyses of human problem solving behavior. Human behavior in different situations depends on the operator's experience and knowledge about the situation and on one's ability to adapt to unfamiliar situations. The categories of human behavior and performance can be expressed in three different levels: skill-, rule-, and knowledge-based behavior.

Skill-based behavior represents sensori-motor performance during acts or activities which, following a statement of an intention, evolve without conscious control as smooth, automated and highly integrated patterns of behavior. Rule-based behavior is typically controlled by a stored rule or procedure which may have been derived empirically during previous occasions or may be developed when needed by conscious problem solving and planning. Knowledge-based behavior is the typical level of behavior in less familiar situations, when no rules for control are available from previous encounters, when problem solving and planning is necessary, when different goals must be considered, and decisions or choices among alternative plans should be taken.

Other than the aforementioned models by Rasmussen, Rouse, and Hunt, there are no directly applicable models of problem solving behavior in coordinating compensation and diagnosis, or for compensation itself. Before presenting such a model, however, the next section considers the role of human expertise and approaches to representing human knowledge. This discussion develops the framework for the computer program which is proposed as a realization of an overall model of human problem solving in dynamic environments.

Representing Human Knowledge

This section reviews a variety of approaches to representing human knowledge in a form suitable for incorporation in a problem solving computer program. Many of the concepts reviewed in this section are embedded in the model proposed in Chapters III and IV.

A fundamental question regarding knowledge representation concerns the tradeoffs between storage and computation. Several models which are similar in essence, have been developed to show how people may trade off between storage and computation (Minsky, 1975, Schank and Abelson, 1977). Minsky's model of frames was one of the first such models.

A frame is a complex data-structure for representing stereotypical situations such as viewing a certain kind of living room or going to a child's birthday party. The frame has slots for the objects that play a role in the stereotypical situation as well as relations between these objects. Attached to each frame are different kinds of information such as how to use the frame, what to do if something happens, default values for slots, etc. As long as nothing too extraordinary appears, the recognition of a frame automatically triggers the appropriate action. A surprise may result when the frame does not include an expected feature or does include an unexpected feature.

A knowledge base is a collection of frames organized in terms of some organizational principles but also looser principles such as the notion of similiarity between two frames. The original frame proposal was basically a framework for developing representation schemes which combined ideas from semantic networks. procedural schemes. linguistics etc. Representation schemes which have adapted the frame proposal are, for example. FRL (Goldstein and Roberts, 1977), KRL (Bobrow and Winograd, 1977), and KLONE (Brachman. 1979).

Scripts, Plans, and Goals as described by Schank and Abelson (1977) are very similar to frames except that they pertain to processes rather than situations. Goals trigger the use of familiar plans which in turn cause the execution of well-practiced scripts.

The most commonly implemented model of knowledge representation is the production system (Newell and Simon, 1972, Shortliffe. 1976, Duda. et al., 1979, Young, 1979). Productions are rules of the form IF <situation> THEN <action>. A network of many interconnected productions is often referred to as an inference network.

Farley (1980) discusses general problematic issues inherent in knowledge-based problem solving. These issues deal with the representation of the environment and general solution plans, the coordination of multiple plan execution and problem solving within an environment over an extended period of time or under time constraints. For example, the environment is represented by the current state, i.e., the current environmental situation, and the goal state, i.e., the situation state which the problem solving system desires the current environmental situation to attain. A problem exists for a problem solving system when the current environmental situation is not a goal state. Problem solving refers to any activity undertaken in the attempt to reduce differences between current and goal state. A general solution plan is the description of a process which is capable of satisfying any of a set of goal states from any of a set of possible current, or initial, states.

In the production system representation of a general solution plan, each plan state serves as the condition part of a rule, whose action part is the operator labeling the link leaving that plan state. A production system is executed by repeatedly selecting a rule whose condition part is satisfied by the current state and executing that rule's action part.

SCHOLAR (Carbonell, 1970) and MYCIN (Shortliffe, 1976) are knowledge-based systems in which a database of information is presented in an associative structure analogous to (but not necessarily exactly like) human structures. While these programs are not models of human performance, they may be thought of as formal qualitative models of the environment in which human performance takes place.

Wesson's (1977) production system description of air traffic control planning is one type of knowledge-based model of human performance. Wesson has built a knowledge-based system that can plan traffic flow, anticipate critical incidents, and issue orders to pilots. In a series of scenarios, the model usually performed better than real controllers.

There seems to be a consensus that expertise is not based solely on the accumulation of facts but also on the development of a problem solving approach. The approach of a problem solver will be heavily dependent upon the internal representation used, and to a lesser extent, dependent upon how the information was originally acquired.

Theories of human problem solving expertise are often not well substantiated by experimental results. The most interesting problem solving tasks are too robust to be controlled experimentally and many experimental tasks are too abstract to provide useful results. For this reason, useful models tend to have narrow applicability whereas broadly applicable models are less directly useful.

Characteristics of Rule-Based Models

In this section the characteristics of rule-based models, especially production systems, are described in more detail, since they form the framework of the model described in Chapter IV.

Rule-based models, especially production systems, have become a fairly popular medium for modeling human problem solving (Waterman and Hayes-Roth, 1978). They excel in flexibility, modularity and expandability (Davis and King, 1977). Within the production system formalism it is possible to express different levels of knowledge as well as different problem solving strategies.

All production systems consist basically of two parts: 1) a set of rules in the form IF <situation> Then <action> or <situation>, and 2) a control structure for administering the rules. The left-hand side of the rules describes a situation to which the rule applies, i.e., a list of things to watch for. The right-hand side describes an action to be taken, i.e., a list of things to do, or information to be gained as a result of employing the rule. A production system is set of productions plus a mechanism to select which one to apply when more than one could be applied. Actions resulting from one production can result in situations that will cause other productions to execute

(Rouse, 1980).

The rules may be either context-specific, i.e., they refer directly to the state of the specific problem, or they may be context-free, i.e., they refer more generally to any problem with a given structure. The control structure, more than the rules themselves, determines how the model behaves.

Another distinction should be drawn between left-hand driven or pattern-directed models and right-hand or goal-driven models. The former selects rules based on the observed situation and the conditions contained in the left-hand portion of the rules. This results in a bottom-up or forward chaining behavior. In the case of situation/action pairs a scheme for conflict resolution is necessary to handle a situation in which more than one action is indicated. This may be accomplished by prioritizing the rules and choosing the first one that matches.

On the other hand, there are several methods of controlling the behavior of the model from the right-hand-side. This results in a hypothesis testing or backward chaining behavior.

Rule-based models seem quite appropriate for modeling the human problem solver in dynamic environments. However, the attributes of rule-based models described so far do not indicate how to model the dynamic aspects of process control. In the next chapter the nature of dynamic processes and the role of the operator in such environments will be discussed. This will lead to a presentation of the general structure of a model of human problem solving.

CHAPTER III

AN OVERALL MODEL

Problem Solving in Dynamic Environments

Most of the well-known rule-based systems such as MYCIN (Shortliffe, 1976) and DENDRAL (Feigenbaum et al., 1971) deal with static problems. Engineering systems, however, are inherently dynamic. For this reason, modeling of human problem solving in the context of engineering systems requires consideration of issues associated with dynamic problems. (For a detailed discussion of the complexity of controlling dynamic processes, see the recent review report by Morris (1982).)

There are different approaches to categorizing the tasks a human operator has to perform when controlling a dynamic system. One is to view the control process in terms of coordination of three goals (Rouse and Morris, 1981a). First the system must be _stabilized_ in the sense of maintaining the state of the system within some allowable range. Second, if possible, system performance should be _optimized_ in order to maximize production, minimize energy consumption, maximize safety etc. Finally, if anything unusual occurs, this event must be _detected_, the source of the unexpected event must be _diagnosed_, and appropriate _compensation_ pursued. These detection, diagnosis, and compensation tasks can be referred to as _problem solving_.

Unfortunately, these three goals cannot be pursued independently. For example, achieving optimal performance may require that one operates the system on the edge of instability. Further, attention devoted to problem solving can result in degradation with respect to stabilization and optimization. Thus, beyond performing the particular tasks associated with stabilization, optimization, and problem solving, one must also achieve an appropriate balance among these three goals.

Thus, the control of an engineering system could be viewed as requiring a two-step process of decision making. First, one must determine which goal is most important at the moment. Then, one must determine the action or actions most appropriate for achieving this goal. This may be quite complicated by the dynamic nature of the system.

Another approach is described by Rasmussen and Lind (1981, 1982). They describe a control system in different hierarchical levels of abstraction, starting with the lowest level, the physical form of a system and then moving up to physical function, generalized function, abstract function and finally the overall purpose of the system. The tasks of the control system will be, by proper action on the system, to ensure that the actual state of the system matches the target state specified by the intended mode of operation. During normal operating conditions, for instance, a set of generic control tasks can be defined: Coordination of functional states in separate units; reconfiguration by switching and valving in order to integrate into higher level functions; adjustment to meet the target state of functions at the next level.

During emergency and major disturbances, an important control decision is to prioritize by selecting the level of abstraction at which the task should be considered. First, judge overall consequences of the disturbance for the plant production and safety in order to see whether the plant mode of operation should be switched to a safer state. Next, consider whether the situation can be counteracted by reconfiguration to use alternative functions and resources. Finally, the root cause of the disturbance is sought to determine how it can be corrected.

The control task can be formulated at the various levels as the activity needed for maintaining or reaching correspondence between a target state and the actual state of the function considered. This task involves identification of the actual state from the measurement of variables related to the physical state of components.

Target states are derived top-down from functional specifications and the reasons behind design decisions, and their determination may be difficult due to the conflicting goals and implicitly given company policies.

Discrepancies between target and actual states determine the task which can be defined at any level. This implies a prioritizing choice considering system dynamics, the nature of disturbances, etc. Finally, the proper sequence of actions is planned from knowledge of the resources available in terms of functions and equipment.

Based on the above lines of reasoning, the model proposed in this thesis assumes that there are four general tasks for the operator in a dynamic environment, two or more of which may have to be performed simultaneously: 1) transition tasks, such as start-up, shut-down, take-off, and landing, 2) steady-state tuning, 3) detection and diagnosis of failures, and 4) compensation for failures. To perform these tasks, the operator has to know: 1) how the process will evolve if left alone, 2) what the effect will be of implementing control actions, and 3) what task is currently appropriate.

In the following subsection these four general tasks are described with emphasis on a modeling view of these tasks, i.e., how to incorporate these tasks into a rule-based model.

Transition Tasks

While the operator performs a transition task, control is often fairly proceduralized although there need not be a formal written procedure.* A certain sequence of actions is often known which will lead to the desired outcome, i.e., the operator knows the goal state. During start-up, for instance, the operator utilizes actions of a known procedure which will lead the actual state of the system to this goal-state. Thus, this is clearly a goal-driven situation and a model for this task should be right-hand driven.

Steady-state Tuning

Steady-state tuning involves actions oriented toward optimizing performance. This calls for a left-hand driven or pattern-directed approach, where the appropriate action depends on a multitude of factors that can only be perceived as a pattern. Procedures are of great value for this task, since the merits and consequences of various approaches may be considered in advance and the one(s) most likely to

* Proceduralized, throughout this thesis, should be understood as informally proceduralized rather than following a written procedure. In many situations humans behave in a proceduralized manner in terms of well-learned scripts of heuristics, and only occasionally are there formal written procedures available for a specific task.

achieve a given set of goals may be selected.

In contrast to transition procedures, tuning procedures are pattern-driven since the operator generally does not work towards a certain goal-state of the system. Instead, the operator applies tuning procedures to maintain the system's state. Thus, a specific pattern, the current situation, triggers an appropriate action.

Failure Detection and Diagnosis

This task will necessarily be performed in parallel to transition and tuning tasks. More specifically, failure detection is active at all times and does not interrupt transition or tuning procedures. Failure diagnosis, however, interrupts the operator while performing a procedure, to which he has to return once the diagnostic task and perhaps any necessary compensation task have been completed. Procedures may not be as valuable for detection and diagnosis because it is rather difficult to anticipate all possible malfunctions.

Thus, rules for failure detection must be monitored during transition tasks, steady-state tuning and compensation for failures. Once an abnormal condition has been detected, then diagnosis may begin. It is possible that the diagnosis task could be either pattern or goal-driven. If the process is not too complex the diagnosis could proceed from the left-hand side. A certain pattern may trigger an appropriate diagnostic action. In a more complex process it may be necessary to establish the integrity of certain critical plant functions in a more structured right-hand driven manner. In this case, planning may be required in terms of the actions likely to lead to a desired goal.

Failure Compensation

Failure compensation would, in most cases, be fairly proceduralized. Once the cause of a disturbance has been diagnosed then, if it is a familiar failure, an appropriate action can be performed. However, if it is an unfamiliar failure, alternative approaches to failure compensation may have to be considered. For example, the human may have to experiment with alternative approaches to compensation.

Summary

The preceeding section describes four general tasks an operator has to perform while he or she is controlling a dynamic process. Transition tasks are fairly proceduralized in that the operator applies a well-known sequence of actions in order to reach a goal. Thus, they are goal-driven. They involve going from one acceptable state to another. Tuning tasks may be proceduralized, but are pattern-driven. The human takes the system from an unacceptable state of the system to an acceptable state following a well-known script evoked by a pattern.

Failure detection and diagnosis are performed in parallel to transition and tuning tasks. There is no procedure to detect a failure but there may be scripts for how to diagnose a failure. This task is goal- or pattern-driven depending on the task complexity. Finally, failure compensation can also be performed by applying learned procedures, however, alternative approaches to compensation are to be considered in unfamiliar situations.

There are some rules that apply to all four tasks and others that apply to only one task (Hunt, 1982). For example, rules for a fairly proceduralized transition task, such as start-up, may be utilized when operations have been cut back during failure diagnosis and have to be restored again. Furthermore, as mentioned above, there are rules for failure detection whose preconditions have to be monitored during all four tasks.

Considering coordination among the four tasks, the human occasionally will proceed hierarchically from goal to subgoal to function to task (i.e., in a linear fashion). More often, however, he or she will skip from task to task and from goal to goal in an opportunistic manner (Hayes-Roth and Hayes-Roth, 1979). The latter reflects a more heterarchical strategy. The proposed model is capable of reflecting both hierarchical and heterarchical behavior.

A Three-Level Model of Human Problem Solving

The first portion of this chapter has outlined the nature of problem solving in dynamic environments typical of engineering systems. The emphasis now shifts to describing how humans deal with problems in these types of environment.

Rouse (1982) has proposed a general and potentially widely applicable model of human problem solving. This model assumes that problem solving occurs at several levels of behavior. It appears that three general levels of problem solving are needed to model human behavior.

1. Recognition and Classification involves detecting that a problem solving situation exists and assigning it to a category. At this highest level the human has to identify the context and category of the problem. The operator first attempts to map the observed state of the problem to an appropriate frame (Minsky, 1975). Failing to recall an appropriate frame, the operator has to classify the situation by its structure, perhaps through analogy to problems with similar structures.

2. Planning is the process whereby the approach to solving a problem is determined. At this level the operator has to

decide upon a course of action. If the situation (i.e., observed <u>state</u>) is familiar, then an appropriate script (Schank and Abelson, 1977) or a standard procedure can be applied. Otherwise, the human must use an approach based on problem <u>structure</u> in order to develop a new plan, e.g., generating alternatives and imagining and valuing consequences.

3. <u>Execution and monitoring</u>. Actual problem solving occurs at this lowest level. Familiar aspects of the current <u>state</u> may be utilized or, if state patterns are not familiar, the execution may rely on features of the <u>structure</u>, in a manner similar to that for the other two levels.

Thus, the model operates on three different levels and on each level with either a state-oriented or a structure-oriented approach depending on the knowledge of the model relative to the patterns displayed. The basic mechanism of this proposed model is such that humans are assumed to have a clear preference for proceeding on the basis of state information rather than using structural information. This reflects an assumed human preference for pattern recognition over more analytical thinking. The basic mechanism of the proposed model is depicted in Figure 1.

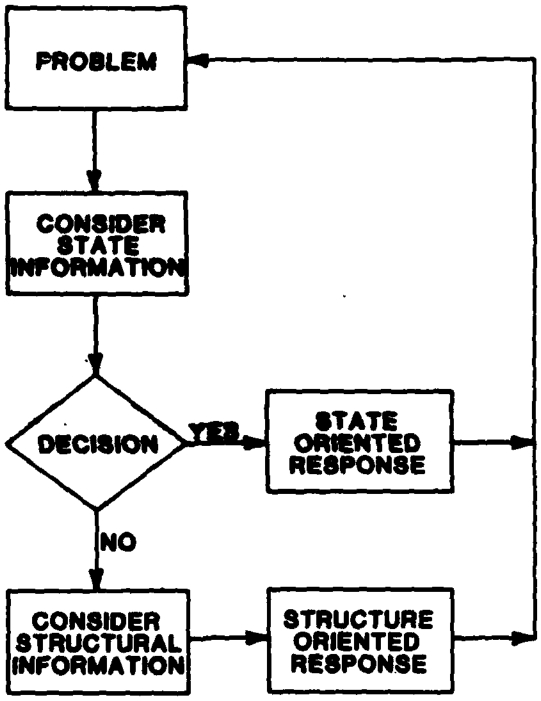


Figure 1. Basic Mechanism of Proposed Model.

In order to realize an operational model of human problem solving based on this general conceptual framework, a control mechanism for the coordination of goals and tasks has to be implemented. Humans often get caught up in the tasks that they are performing and lose sight of their goals. In addition, because of the dynamic nature of the task, goals may be preempted, temporarily or permanently, before they are reached. The next chapter describes a possible control structure that allows representation of these phenomena.

CHAPTER IV

REALIZATION OF A RULE-BASED MODEL

In the preceding chapter a general model of human problem solving was discussed. The development of this model, in the form of a rule-based computer program, has been the goal of this thesis. While the rules which the model utilizes naturally depend, to a great extent, on the specific problem to be solved, one particularly important goal of this research has been to give the model a generally applicable structure. With such a structure the model should be easily adjustable to different dynamic problem solving environments.

Structure of KARL

KARL (Knowledgeable Application of Rule-based Logic) is a rule-based computer program, a model which consists of a set of production rules that comprise the knowledge base and a control structure that accesses the knowledge base. The model contains approximately 180 production rules including simple and complex rules. The condition parts of complex rules contain several logical expressions. Of these 180 rules, approximately 140 pertain specifically to the process control task where KARL was evaluated; obviously, this set would change for application to other tasks.

A simplified flow-chart of KARL is given in Figure 2. The production rules are embedded in a framework, which represents the four tasks associated with controlling a dynamic process, as well as the three levels a human generally operates on while controlling a dynamic process. The control structure consists of two modules: 1) a pre-processing module. where the necessary information about the system's state is processed in order to identify the current task the model is performing (CURRENT-TASK), and 2) a branching mechanism that accesses the portion of the knowledge base containing the rules the model requires in order to perform the current task (CONTROL). The model assumes that whatever task it was last doing it is still doing, i.e., it does not re-identify the current task each time. It changes the task when the perceived information forces it to do so.

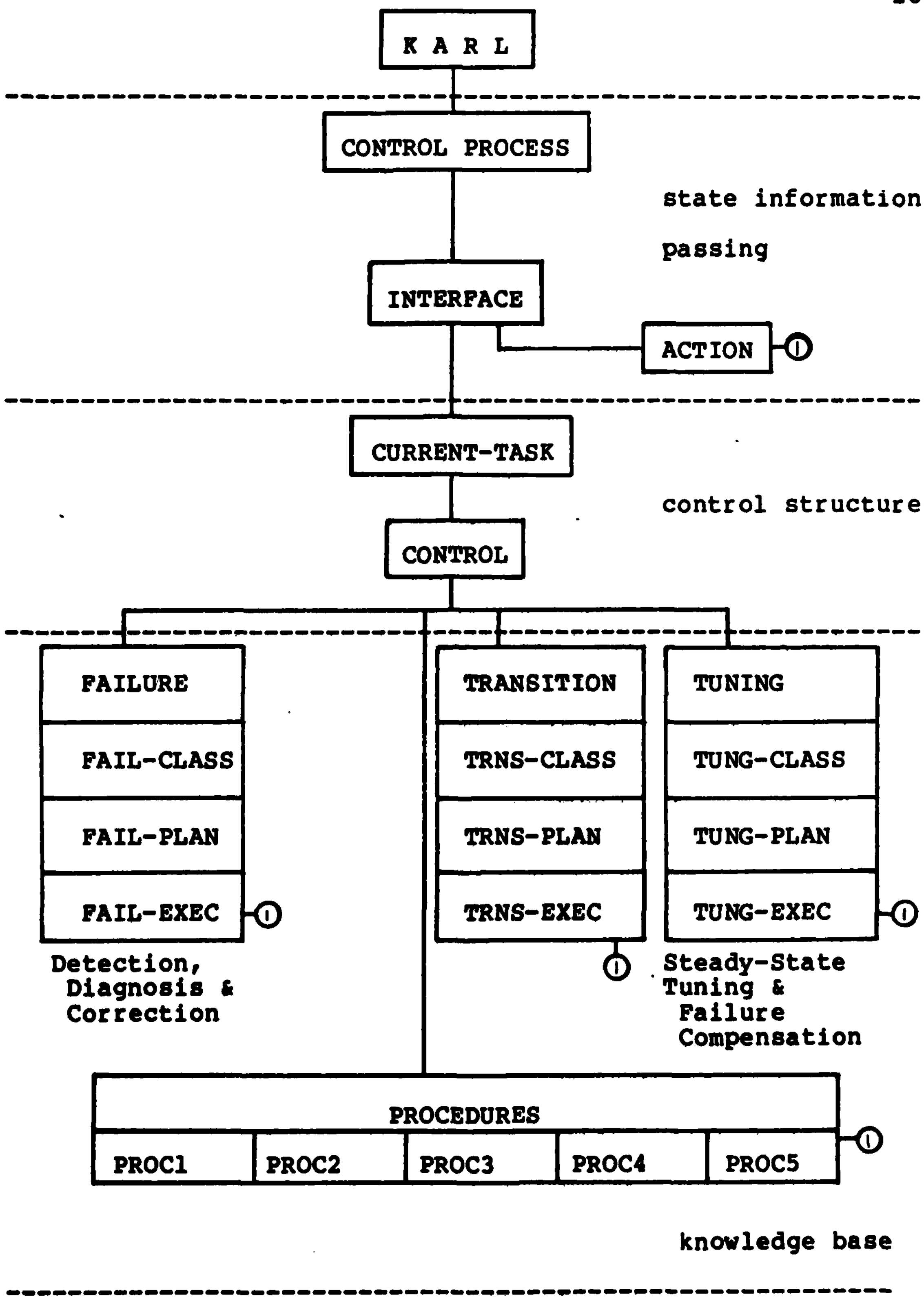


Figure 2. Flow-chart of KARL.

The knowledge base consists of four subsets, each of which is associated with one of the four general tasks although these subsets deviate somewhat from the tasks discussed earlier. FAILURE contains rules for detection, diagnosis and correction of failures. TRANSITION contains a rather proceduralized sequence of rules. TUNING contains rules for normal operating conditions and failure compensation. Finally, PROCEDURES contains standard sequences of rules, each of which is applicable to particular operating situations.

It is necessary to make clear that there are two kinds of procedures. Transition procedures are generalized sequences of actions which are located in the TRANSITION part of the model. As mentioned before these procedures are goal-driven. They differ from Tuning procedures in that they have to be performed in order to control a system. Further, they occur, in general, at well known specific points of time. An example of a transition procedure is landing an airplane. The goal, landing, can only be reached by applying a well-learned procedure that prescribes to the pilot what sequence of actions are necessary in order to land the plane.

Tuning procedures, however, are sequences of actions, often well known also, but not always necessary in order to reach a goal. Flying an airplane the pilot may receive an announcement that the runway has to be temporarily closed due to snow removal. Pilots are generally fairly familiar with such situations and deal with those abnormal situations in fairly proceduralized manners. They either will enter a holding pattern or cruise to an alternate airport. This additional procedure, during navigation of an airplane, may or may not occur as one proceeds to the goal.

PROCEDURES includes those procedures which are applied, if the situation requires them, at unforeseen points of time, but are not necessarily applied in order to fulfill the overall goal. For the process control discussed later, these procedures describe situations where the system state requires a certain action sequence in order to restore stable system operation.

Each module, except for PROCEDURES, is divided into the three levels observable in human problem solving behavior. After the classification of a situation (CLASS), different possible actions are evaluated (PLANning), from which an appropriate one is then executed (EXEC).

Functional Aspects of KARL

Referring to Figure 2 the information flow through KARL can be explained. The control process passes information about the system's state to the model and prompts it for an action command. In INTERFACE this state information is converted into a form suitable for the rules embedded in the knowledge base. In CURRENT-TASK the model determines the current task by means of current state information and, in particular, by knowledge about the system gained from previous state information acquired and subsequent actions. At this point, CONTROL branches to one of the four modules in the knowledge base.

As mentioned above, the model works both hierarchically and heterarchically. The rules in FAILURE, TRANSITION and TUNING have been constructed in such a way that, with the starting point at the top (CLASS), either a rule in a lower level of operation is invoked (i.e., from CLASS to PLAN, from CLASS to EXEC or from PLAN to EXEC), or a rule in another task is invoked (e.g., from TRNS-CLASS to FAIL-CLASS or from TUNG-EXEC to PROCEDURES).

PROCEDURES is not divided into these three levels, since it is assumed that for a given situation the commands to be given are specified until the system is stabilized. This does not mean, however, that a predetermined command sequence is given. Applying a procedure means that basic actions are known that will return the system to stable operation. These are determined by invoking rules which contain knowledge about which command is to be given when. However, while the model is following a procedure, a failure may occur so that the procedure may be preempted in order to perform failure diagnosis and correction. While failure diagnosis and correction take place, CURRENT-TASK and CONTROL take care of "remembering" the current operating task.

In order to solve conflict situations, i.e., when the condition parts of two rules or more are satisfied, the following priority structure is embedded in the knowledge base. The production rules are rank-ordered, i.e., the rule with the highest priority is tested first. If this rule matches the situation no other rule is tested. If the condition part of this rule does not match the situation, the next rule is tested, and so on. In order to change priority, the order of the rules has to be changed, or a different mechanism to select rules has to be implemented. For example, a fuzzy selection as suggested by Hunt and Rouse (1982) might be employed.

Summary

The essential elements of KARL include the four general tasks an operator has to perform while he or she is controlling a dynamic system. Each task is divided into the three levels of problem solving behavior: classification, planning and execution. Throughout this structure the model works both hierarchically, i.e., going down from classification to planning to execution within tasks, and heterarchically, i.e., going across between tasks.

It should be noted that the proposed model is not intended to be an "Expert System" such as MYCIN (Shortliffe, 1976). MYCIN has been designed to diagnose and prescribe treatment for real medical conditions to the best of its artificially intelligent ability. Towards this end the designers have combined human reasoning abilities with the rapid retrieval and calculating abilities of the computer to create a problem solver that often out-performs its human counterparts.

However, the goal for KARL is not that it should out-perform humans but that it should match human performance, both good and bad. A model that accurately represents both the efficient and inefficient elements of human performance would be of great value in the design of decision aids and/or development of effective training programs. In the next chapter the proposed rule-based model of human problem solving is applied to a process control simulation.

CHAPTER V

PERFORMANCE OF KARL
IN A PROCESS CONTROL TASK

The first test of the proposed model, KARL, was performed in a process control simulation that was developed for studying human problem solving. This simulation environment was chosen because of the availability of extensive data on human problem solving in this simulator.

PLANT.

PLANT (Production Levels And Network Troubleshooting) is an abstract computer-based simulation of a continuous, fluid processing plant through which generic raw material is transformed into generic finished product. The PLANT operator's task is to supervise the flow of fluid through a series of tanks interconnected by valves so as to produce an unspecified product. The operator's goal is to maximize production given the "physical" limitations of the system (such as tank or valve capacity or reliability of system components).

Tanks are organized in columns, where tanks in the left-most column receive input and tanks in the right-most column produce output. In general, the flow through the system is from left to right, and any pair of connected tanks behaves as a second order system.

Tanks are connected by valves, which may be opened and closed by the operator via appropriate commands. Furthermore, the operator controls the input to and output from PLANT by specifying the number of units of fluid per tank to be pumped in and pumped out, respectively. A sample network of the PLANT simulation is depicted in Figure 3.

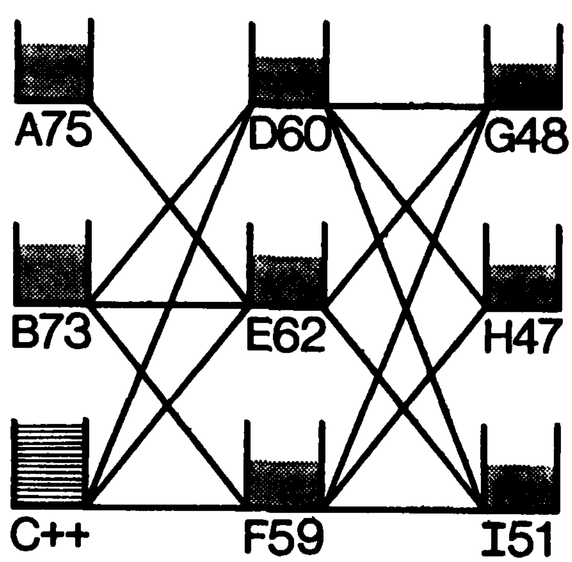


Figure 3. Example PLANT Network

Since pumps, valves, and tanks, may fail in different ways, several diagnostic commands are available to the operator. Furthermore, there are commands for failure compensation, such as sending a "repair crew" to the site of the failure. (See APPENDIX A for a list of all commands.)

Safeguards incorporated into the system inhibit loss of control of the system and prevent system damage. For example, valves are closed automatically, i.e., tripped, when the amount of fluid flowing through them is too great. The operator can recover from trips by reducing input and output and diligently reopening valves until flow is stabilized. For more detailed information about PLANT see Fath (1982) and Morris (1983).

Assessing the Model Performance

The ultimate goal for any model of human behavior is to duplicate the behavior of the human being modeled (Hunt, 1981). The degree of success obtained is highly dependent upon the measures used to evaluate the performance of the model. One might develop a model for solving problems such that it matches human performance with PLANT very well in terms of overall production obtained. However, the model might use strategies completely different from those of humans. Since the goal of the proposed model is to match human behavior in terms of problem solving approaches, rather than to match only the final problem results, emphasis was placed on the sequence of actions taken by the model in comparison to the subjects.

A large quantity of data was available from experiments performed by Morris (1983). A sample of this data was used to develop the rules of the model and to evaluate the resulting model behavior and performance. Specifically, data for all 32 subjects in sessions 8 through 11 of the experiment of Morris (1983) was utilized for the comparison reported here.

Comparison with Subjects

An overall evaluation of the model was performed by comparing the total production achieved by the subjects and by the model. Furthermore the number of system trips and number of alarms, which may be seen as measures of stability, were compared. In order to compare the actions of the model and those of the subjects, the model was used to select the action it would have made in each situation that the human subject viewed. Then the action actually taken by the subject was read from a data file. The actions for the model and the subjects were then recorded in a data file for later analysis. Finally, the problem was updated with the action selected by the subject. This process was repeated until the problem was solved.

The subject's choice was used for updating since, otherwise, any deviation of the command sequences of model and subjects would have resulted in divergent system states and inherently different command sequences. To avoid this problem the model always selected actions according to the situation viewed by the subject. This was assured by always implementing the subject's choice.

Before discussing the performance of the model, information about the experiment performed by Morris (1983) as far as necessary for understanding the model application is briefly outlined. Morris was interested in the effects of level of knowledge upon human problem solving in a process control task. Her experiment involved 32 subjects, who were divided into four groups each of which received a different set of instructions.

Group A received minimal instructions which were directed at _what_ kind of system is it, i.e., information about the concept of the plant, discussion of the goals of PLANT operations, operational constraints, possible malfunctions, and command options available. Group B received the same set of instructions and, in addition, information as to _why_ the system should be controlled in a certain manner, i.e., principles of the plant in terms of dynamic interrelationships. Group C received minimal instructions and a set of procedures, i.e., _how_ the system should be controlled in both general and more specific forms. Finally, group D received an instruction set containing principles and procedures, as well as relationships between principles and procedures. These relationships were more directly related to the "whys" of PLANT operations; here the rationale behind the information in the procedures was presented in terms of concepts discussed in the principles.

Overall Measures

Total Production

To compare the model with the subjects on total production the model was allowed to run by itself, without being updated using the subject's actions. It should be noted that production was never used by the model to make a decision. Certainly, the model uses production as an overall goal as the subjects, but it does not use production at any instant to drive rules. Table 1 shows the total average production for KARL and for all 32 subjects from the four instructional groups in the four production runs (i.e. sessions 8 through 11 of Morris' experiment).

Morris (1983) found that there was no significant difference in total production achieved among the four instructional groups. However, there was wide variability within groups. It can be seen that the model succeeded very well in matching the average or typical performance for all four groups.

Table 1. Total Production
Comparison of Subjects and KARL.

Groups	Production runs Average production:			
	1	2	3	4
A	147,715.9	135,765.8	164,298.4	133,224.4
B	145,872.8	139,839.4	166,433.3	136,160.6
C	174,725.6	148,066.5	167,032.9	140,749.9
D	161,398.9	146,911.5	171,959.3	146,989.9
Mean	157,428.3	142,645.8	167,430.9	139,281.2
KARL	156,960.0	144,690.0	168,480.0	151,710.0

Stability

Another performance measure is the overall stability of the system. This can be indirectly measured by system trips and alarms. Valves of the system were tripped, i.e., were closed automatically by the system's safety system, when the magnitude of the flow through them was too great. One way this could occur was when valves were opened between tanks whose differences in fluid levels were too great, creating a pressure difference that led to excessive flows. Thus, large differences in levels were to be avoided. Warning alarms were given when a tank reached its maximum capacity or the fluid in it dropped to zero. Thus, a stable system is characterized by approximately equal fluid levels in tanks.

Table 2 shows the total average number of system trips and total average number of alarms for KARL and the 32 subjects in the four groups and in the same four production runs as noted in the preceding section. Morris found significant differences in system stability between those groups who received procedures, i.e. how to work the system (i.e. Groups C and D), and those who did not (i.e. Groups A and B). Since the rules in the model were developed according to those procedures, it is not surprising that the model matched the instructional groups with procedures better than the others.

The number of alarms of the model is significantly lower than the average of all four groups. That is due to a general rule in the model to reduce input and output to zero whenever fluid levels exceeded critical values. Subjects, however, continued to try to produce even when critical fluid levels were exceeded.

Table 2. Stability Measures
Comparison of System Trips and Warning Alarms of Subjects and KARL.

Groups	Production runs Average Number of System Trips and Alarms:			
	1	2	3	4
	Trips/Alarms	Trips/Alarms	Trips/Alarms	Trips/Alarms
A	565.7/93.75	507.1/93.00	343.4/12.50	497.6/73.25
B	544.8/40.25	433.0/41.75	318.1/ 9.63	397.5/55.50
C	374.4/ 1.00	353.0/ 9.25	285.9/ 1.75	307.4/18.00
D	321.8/ 1.88	333.4/ 6.25	224.6/ 1.88	261.5/14.75
Mean	451.7/34.22	406.6/37.56	293.0/ 6.44	366.0/40.38
KARL	259.0/ 0.0	308.0/ 0.0	235.0/ 0.0	270.0/11.0

Action-by-Action

As emphasized earlier, an interesting aspect of evaluating a model is the comparison of how the model and subjects reached their final state, i.e., comparing the sequences of actions. In order to study the strategies of the subjects the model was used to determine what action it would have chosen in each situation seen by the subject. Action-by-Action comparisons were performed for all 32 subjects in the four previously examined production runs.

Levels of Agreement

Since there are 16 different commands available to operate the system* and an almost infinite number of arguments for the commands it seemed reasonable to evaluate KARL in terms of several possible levels of agreement between each subject and model.

The highest level of agreement occurred when subject and model employed the same command plus same argument (Level I). It cannot expected, however, to achieve a high percentage of matching in this category. For example, for the "p" command, i.e., pump in or pump out a certain amount of fluid (see APPENDIX A), there are infinite possibilities for the specified amount, although only a limited number is reasonable. Thus, the model was restricted to eight values, i.e., 0, 50, 100, 150, 200, 210, 220, and 230 units. However, subjects were free to choose any amount of input or output.

The next highest level of agreement occurred when subject and model employed the same command, but not necessarily the same argument (Level II). On this level a reasonable percentage of matching can be achieved. However, from the 16 commands available, there are some commands

* APPENDIX A contains a listing of all possible commands available to the PLANT operator.

which were never used by the model, e.g. the "ct" command, i.e., "close all valves from any tank". Since an overall policy for the system was "keep all valves open", the model contained no rule to close all valves from one tank. Only on very rare occasions would the model close individual valves, i.e., use the "cv" command.

The next level of agreement occurred when subject and model employed the same type of command (Level III).* There are six different types of commands, and agreement at this level can be interpreted as implying that subjects used the same or similar rules to choose their actions as the model.

The final level of agreement included Levels I, II, and III and, further, those agreements which happened over two iterations (Level IV). For example, if the subject utilized a "po" command and then an "ot" command, and the model utilized the same commands but in opposite order, these agreements were included in this category.

Table 3 shows how the model compared to all subjects in four production runs. On Level I an average agreement of 17.67 % was found, on Level II 34.83 %, on Level III 52.85 % and on Level IV 60.46 %. No significant differences were found between the four groups on any level of agreement.

* APPENDIX B contains a listing of the different types of commands.

This is somewhat surprising since the model's actions were chosen according to proceduralized rules. and it was expected that the groups who received procedures (i.e., Groups C and D) would have been matched better than those who did not (i.e, Groups A and B).

Table 3. Levels of Agreement of Subjects and KARL.

Groups	Levels of Agreement			
	I	II	III	IV
A	16.70	33.18	53.90	61.37
min	1.00	5.80	45.20	52.60
max	35.10	57.10	65.60	79.40
B	14.93	32.95	50.91	59.12
min	5.60	17.00	39.40	44.10
max	26.20	59.30	70.90	86.40
C	21.54	36.80	54.40	61.09
min	9.20	18.00	40.00	48.10
max	32.70	57.70	64.90	73.20
D	17.50	36.39	52.21	60.26
min	5.00	14.20	35.60	40.00
max	32.20	57.50	65.60	83.50
Mean	17.67	34.83	52.85	60.46

Within groups there was wide variability among subje::s and it was found that the model agreed more with subjects who had fewer trips and alarms. i.e., a more stable system, than those subjects who had many trips and alarms. Table 4 shows the degree of agreement of eight subjects. a good and a bad subject of each group, with the model. System trips and alarms of these subjects are depicted in the left-most column and agreements in percent in the other columns. Morris (1983) found significant differences in system stability between procedures and no-procedures groups. However, since both groups included good and bad subjects, the difference in agreements were not sufficient to produce significant differences between groups.

Table 4. Levels of Agreement of Subjects and KARL in Comparison to System Stability.

Groups	Levels of Agreement			
	I	II	III	IV
Trips/ Alarms*				
A				
g: 144/0	21.03	39.82	47.43	62.86
b:1038/478	1.00	6.20	52.00	57.40
B				
g: 174/0	22.82	59.28	70.92	86.35
b: 614/0	6.40	17.20	44.20	47.00
C				
g: 156/0	21.03	47.20	59.28	69.57
b: 577/0	19.09	34.20	52.13	59.96
D				
g: 153/10	19.91	57.49	65.55	83.45
b: 402/0	10.96	26.17	50.34	54.13

* The numbers in this column correspond to the number of system trips and number of alarms of a good (g) and a bad (b) subject in each group.

There are some rules within KARL which are not used by some subjects. and in those cases the model failed to match the subject's choice. However, the goal was to avoid making every action a special case with its own rule. Although eventual plans for KARL involve changing the production rules of the model in order to match poorer subjects or subjects with limited knowledge. it is clear from the above results that it is not possible to develop a single set of rules which agrees with subjects' choices as a group; there is simply too much within group variation.

Causes of mismatches

Since an overall match of 60.5 % was found, it was of interest to analyze what kind of mismatches occurred. To categorize these mismatches, a matrix of percentages of commands given by subjects vs. those given by KARL was constructed. The matrix containing the Level III comparisons, i.e., same type of command, is depicted in Table 5. The sum of the diagonal entries (i.e., 48.5 %) represents the percentage of agreement on Level III.

Further, comparisons involving unimportant differences were included in this category. These are uses of command-type "a", i.e., "check flow from any tank", and of command-type "s", i.e., "skip iteration". Very often, the situation did not justify the use of the "a" command.

which should be used as a diagnostic action when discrepancies from expected fluid levels call for it. Disagreements in the use of these commands, i.e., KARL used "a" or "s" commands and subjects used the opposite in unjustified situations, were included in Level III comparisons. Including this type of unimportant differences, an agreement of 52.85 % was found (see also Table 3).

Table 5. Distribution of Commands
Used by Subjects and KARL.

		KARL					
		1	2	3	4	5	6
Subjects		p	s	o	c	a	r
1	p	4.62	1.01	3.58	0.12	2.58	0.05
2	s	3.00	1.14	2.83	0.19	3.75	0.07
3	o	14.12	0.89	34.07	0.02	0.46	0.13
4	c	0.76	0.18	1.74	0.02	0.31	0.02
5	a	7.56	2.03	4.53	0.31	8.46	0.18
6	r	0.34	0.14	0.47	0.01	0.13	0.10

Table 6. Percentage of Commands
Used by Subjects Given a Specific
Command by KARL.

		KARL					
		1	2	3	4	5	6
Subjects		p	s	o	c	a	r
1	p	15.20	18.72	7.59	17.50	16.45	9.73
2	s	9.88	21.24	5.99	28.06	23.89	13.42
3	o	46.43	16.47	72.16	2.78	2.93	23.42
4	c	2.51	3.38	3.68	3.33	1.97	3.02
5	a	24.86	37.68	9.60	47.22	53.95	32.55
6	r	1.12	2.52	0.99	1.11	0.82	18.12

While Table 5 shows the percent usage of each of the six types of command, Table 6 shows the percentages of the subject's choice given a particular type of command by KARL. Table 6 is simply a transformation of Table 5. In Table 5 the entry at the intersection of subject's "o" command and KARL's "p" command reads: in 14.12 % of all action choices the subjects used the "o" command, when KARL used the "p" command. In Table 6 the corresponding entry has been computed by dividing the entry of Table 5 by the sum of the column, i.e., in this example, the percent usage of "p" commands by KARL. Thus, the corresponding entry in Table 6 reads: given KARL's "p" command, subjects used the "o" command 46.43 % of the time.

The percentage of disagreements for all comparisons was 39.5 %. The most important causes of mismatches are summarized below:

1. Given the "p" command by KARL, 46.4 % of the subjects' choices were the "o" command. This disparity accounts for an overall 35.7 % (i.e., 14.1 % of all comparisons)* of the mismatches. The highest priority of the model is to adjust input and output appropriately before utilizing any other command. While the model running by itself did not use the "p" command very often, most

* The information in parentheses gives the reader the possibility to refer to Table 5.

subjects did not adjust input and output as the model would do. Thus, upon viewing the state of PLANT after the subject's command had been implemented, the model continued to propose the "p" command, which is the reason for this mismatch.

2. Similarly, given the "p" command by KARL, 24.9 % of subjects' choices were the "a" command, i.e., "check flows". The subjects preferred to check flows rather than adjust input/output. This difference in priorities contributed 19.2 % (i.e., 7.6 % of all comparisons) to the overall mismatch.

3. Finally, given the model's "p" command, a further 9.9 % of the subjects' choices were the "s" command. This contributed 7.8 % (i.e., 3.0 % of all comparisons) to the overall mismatch.

It can be seen from 1. through 3. that the highest percentage of mismatches (i.e., 62.7 %) resulted from subjects' choices of input and output. Even subjects who received procedures which prescribed to them how to adjust input/output in certain situations, did not fully utilize this information.

4. Subjects utilized the "a" command 9.6 % of the time while the model gave the "o" command. i.e., "open valves". The second highest priority rule for the model was "to keep all valves open" before checking any flow. This contributed 11.4 % (i.e., 4.5 % of all comparisons) to the overall mismatch.

5. Similarly, subjects gave the "s" command 5.9 % of the time while the model. due to the priority noted in 4., gave the "o" command. This disagreement contributed 7.1 % (i.e. 2.8 % of all comparisons) to the overall mismatch.

6. Given the model's "o" command, 7.6 % of the time subjects used the "p" command. It was found that the subjects often acted conservatively, i.e., they decreased input/output although the system's state did not call for it. This contributed 9.1 % (i.e., 3.6 % of all comparisons) to the overall mismatch.

7. Finally, given the model's "o" command, 3.7 % of the time the subjects used the "c" command. This is an indication that subjects did, in general, much more fine tuning than the model. The model was instructed "to keep all valves open" rather than to close valves in order to adjust the fluid levels. This contributed 4.3 % (i.e., 1.7 % of all comparisons) to the overall

mismatch.

The mismatches described above account for 94.6 % (i.e., 37.4 % of all comparisons) disagreements. All other types of mismatches contributed less than 1 %.

Summary

The preceding sections described the results of various comparisons of subjects' behavior and that of the model. The overall production comparison yielded a very good match of the average production for all subjects. This agrees with the findings of Morris (1983) in that the four groups did not differ in average production. However, system stability differed for the four groups depending on instructions received, and since the model incorporated in its rules the procedures which some subjects received, it agreed with those subjects better in terms of system trips and alarms.

The action-by-action comparisons yielded no significant differences between the four groups, which was due to wide variability among subjects' action sequences within groups. A higher level of agreement could be achieved by adjusting the model to individual subjects, both in terms of rules and priorities.

Table 7 shows the percentage of all types of commands of the four groups and of the model when it ran by itself. It can be seen that the model agrees better with procedure groups (i.e., groups C and D) in terms of "s", "o" and "c" commands than with no-procedure groups (i.e., groups A and B). However, performing action-by-action comparisons the model had to adjust to the situation viewed by the subjects, and so the differences could not appear.

The "a" command was used by the model significantly less than by the subjects because the model used this command more diagnostically than the subjects who seemed to use it very often just to spend time (see also the discussion of unimportant differences in the last section). This may have been due to the fact that the model contains a first order approximation of the second order PLANT dynamics and was able to predict expected fluid levels. KARL checks the flow if the fluid level in the tanks deviates from the expected level, i.e., the predicted level. Thus. failure detection is based on violation of expectations.

Table 7. Comparisons of Types of Commands of Subjects and KARL.

Groups:	A	B	C	D	KARL
command					
"p"	14.50	8.91	12.6	11.89	24.95
"s"	6.02	9.83	15.76	12.42	13.31
"o"	54.46	52.14	46.79	44.78	42.06
"c"	3.96	4.50	1.69	1.81	1.56
"a"	19.82	23.36	21.94	27.66	16.15
"r"	1.19	1.11	1.17	1.27	0.97

It was found that some rules within KARL followed the PLANT instructions too strictly to match several subjects. For example, the model contains a rule which proposes that if the system is stable, i.e., all valves are opened and the height differences are within a specified range, then output and input should be some high, optimal value. However, some subjects appear to be conservative and do not like to bring the system to its limits. In these cases the model continues to give the action command for input and output, but the subject never goes that far.

Similarly, if the system is destabilized, the model wants to reduce input and output in order to restore a stable system. Subjects, however, since their main goal is to achieve high production at all costs, continue to produce even if it would be better to restore system stability. This behavior was observed very often and resulted in degradation of the match. It might be reasonable to inhibit the model's excessive use of the "p" command by allowing it just one "p" command per N iterations or one "p" per N operator actions. This would possibly improve the match.

Furthermore, the model's overall strategy, "keep all valves open", was not followed by several subjects, even when they were instructed to do so. Subjects opened and closed individual valves very frequently, a behavior which was not implemented in the model at all. Thus, subjects did much more fine tuning than the model.

These results lead to several interesting behavioral interpretations. First, not all subjects appear to know, or at least utilize, all of the information provided in their instructions, despite the fact that a written examination, administered by Morris (1983), indicated subjects had learned their instructions. Second, some subjects appear to be conservative in terms of not operating the system as close to the limits of production as is possible, and tune the process much more than necessary. These results suggest some interesting avenues for training and/or aiding of operators.

CHAPTER VI

SUMMARY AND CONCLUSIONS

This thesis has discussed an approach to modeling human problem solving in dynamic environments. A variety of models of human problem solving were reviewed, and it was concluded that there is a need for a general model of human problem solving in dynamic environments. Several approaches to representing human knowledge in a form suitable for incorporation in a problem solving computer program were discussed. This leads to a discussion of a three-level model of human problem solving.

This model was realized in a particular rule-based computer program. This program was discussed in detail with emphasis on the general structure of the model which enables adjustment (via the rule base) to different problem solving tasks. This model was applied to a process simulation task and the results were presented and discussed. The model matched the subjects performing this process control task quite well in terms of overall performance, and reasonably well in terms of the sequence of actions utilized by subjects, i.e., <u>how</u> the subjects reached their goal.

There are several limitations in the model which will be the subject of further research. The rules in the model were developed according to procedures which were found appropriate to operate the system. Thus, for almost every situation the model utilized a script which triggered an appropriate sequence of actions. If there was no script available the model chose a tuning action which might or might not be appropriate. Although "planning" was incorporated in the framework of the model, the model did not actually plan what action to give and when to give it. Thus, the planning aspect of the model needs considerable elaboration.

An interesting conclusion emerged from the action-by-action comparisons. It was expected that subjects who received procedures should agree with the model better than subjects who did not. However, large individual differences resulted in no statistically significant difference being found. It can be concluded that many subjects who received inherently rule-based training, i.e., procedures. did not follow these instructions. In the written test, however, these subjects answered questions, which asked for the appropriate procedure to use in certain situations, according to the training documentation they received, i.e., they learned their instructions. Thus, the action-by-action comparison implies that "experts" do not always tell what they actually do. It is not asserted that subjects after a number of training sessions are experts. This notion is used here to show that the subjects knew what to do, as a written examination indicated, but did not always do it. This may explain why there were no differences found in matching subjects of different groups.

Changing the knowledge base of the model both in terms of rules and priorities should take into account the fact that procedures provided as training documentation do not necessarily describe subjects' behavior. Another point which will be considered in view of the above observations is the model's potential for making errors in the same way and for the same reasons as humans.

In summary, the model is throughout fairly proceduralized, and further research will involve making the model more general in terms of planning and reasoning about actions. Training procedures are not necessarily followed by subjects, so the model should be allowed to deviate from learned procedures. Another idea is to implement the model as an online method for testing the subjects' use of instructions. Finally, the model's usefulness will also be tested in different problem environments.

APPENDIX A

PLANT COMMAND OPTIONS *

As the operator, you have the following commands available:

ovI,J	Open the valve between tanks I and J
cvI,J	Close the valve between tanks I and J
ocK	Open one valve per tank in column K
ccK	Close one valve per tank in column K
otI	Open all valves from tank I
ctI	Close all valves from tank I
piN	Set input per input tank to N units
poN	Set output per output tank to N units
skN	Skip N iterations; the system will be updated N times before the display is updated
flI,J	Check the flow from tank I to tank J
afI	Check all flows from tank I
rvI,J	Repair the valve between tanks I and J
rpI	Repair the pump associated with tank I
rtI	Repair the rupture of tank I
rs	Repair the PLANT safety system
st	System trip; close all valves and stop all input and output

* Taken from Morris (1983)

APPENDIX B

PLANT COMMAND TYPES

"p"	Set input or output
"s"	Skip iteration(s)
"o"	Open valve(s)
"c"	Close valve(s)
"a"	Check flow(s)
"r"	Repair valve, pump or tank

REFERENCES

Bobrow, D. and Winograd, T. An overview of KRL, a Knowledge Representation Language. Cognitive Science, Vol. 1, No. 1, January 1977.

Brachman, R. On the epistomological status of semantic networks. In Findler, N.V. (Ed.) Associative networks: Representation and use of knowledge by computer, MIT Press, 1979.

Carbonell, J.R. AI in CAI: An artificial intelligence approach to computer assisted instruction. IEEE Transactions on Man-Machine Systems. MMS-11, No. 4, 1970.

Curry, R.E. A model of human fault detection for complex dynamic processes. In J. Rasmussen and W.B. Rouse (Eds.) Human detection and diagnosis of system failures. New York, Plenum, 1981.

Davis, R. and King J. An overview of production systems. In E.W. Elcock and D. Michie (Eds.) Machine Intelligence 8: Machine Representation, 1977.

Duda, R., Gaschnig, J. and Hart, P. Model design in the prospector consultant system for mineral exploration. In D. Michie (Ed) Expert systems in the microelectronic age, Edinburgh University Press, 1979, pp. 153-167.

Fath, J.L. Design and evaluation of a process plant simulator. M.S.I.E. Thesis, Georgia Institute of Technology, Atlanta, Georgia, September, 1982.

Farley, A. Issues in knowledge-based problem solving. IEEE Trans. Sys., Man, and Cybern. Vol. SMC-10, No. 8, 1980, pp. 446-459.

Feigenbaum, F.A., Buchanan, B.G., and Lederberg, J. On generality and problem solving: A case study using the DENDRAL program. Machine Intelligence, Edinburgh University Press, 1971.

Goldstein, I. and Roberts, R.B. NUDGE: A knowledge-based scheduling program. Proceedings IJCAI-77, Cambridge, Mass., August 1977.

Greenstein. J.S. and Rouse W.B. A model of human decision making in multiple process monitoring situations. I*EEE Trans. Syst. Man. Cybern.*, vol. SMC-12,1982.

Hayes, J.R. On the function of visual imagery in elementary mathematics. In W. G. Chase (Ed.) V*isual information processing*, New York, Academic Press, 1973.

Hayes-Roth, B. and Hayes-Roth, J. A cognitive model of planning. *Cognitive Science*, Vol. No. 4, 1979.

Hunt, R.M. Human pattern recognition and information seeking in simulated fault diagnosis tasks. Ph.D. Thesis, University of Illinois at Urbana-Champaign, 1981.

Hunt, R.M. Rule-based models of human control of dynamic processes. *Proceeding of the International Conference on Cybernetics and Society*, Seattle. 1982, pp. 686-689.

Hunt, R.M. and Rouse, W.B. A fuzzy rule-based model of human problem solving. In *Proc. of the American Control Conference*, Arlington, Virginia, June, 1982.

Minsky, M. A framework for representing knowledge. In P. H. Winston (Ed.) *The psychology of computer vision.* New York: McGraw-Hill, 1975.

Morris, N.M. Human problem solving in process control. Ph.D. Thesis, University of Illinois at Urbana-Champaign, 1983.

Morris, N.M. The human operator in process control: A review and evaluation of the literature. Technical Report No. 82-1, Center for Man-Machine Systems Research, Georgia Institute of Technology, Atlanta, Georgia, 1982.

Newell, A. and Simon, H.A. *Human problem solving*. Englewood Cliffs. New Jersey, Prentice Hall, 1972.

Niemela, R.J. and Krendel, E.S. Detection of change in plant dynamics in a man-machine system. *IEEE Trans. Sys., Man. and Cybern.*, Vol. SMC-5, 1975, pp. 615-617.

Rasmussen, J. *Notes on diagnostic strategies in process plant environment.* Riso National Laboratory, Roskilde, Denmark, Report M-1983, January, 1978.

Rasmussen, J. The human as a system's component. In H.T. Smith and T.H.G. Green (Eds.) Human interactions with computer, London, Academic Press, 1980.

Rasmussen, J. Models of mental strategies in process plant diagnosis. In J. Rasmussen and W.B. Rouse (Eds.) Human detection and diagnosis of system failures, Plenum Press, New York, 1981, pp. 241-258.

Rasmussen, J. and Lind, M. A model of human decision making in complex systems and its use for design of system control strategies. In Proc. of the American Control Conference, Arlington, Virginia, June, 1982.

Rasmussen, J. and Lind, M. Coping with complexity. In Proc. of the First European Annual Conference on Human Decision Making and Manual Control, Delft, The Netherlands, 1981.

Rouse, W.B. Systems engineering models of human machine interaction, North Holland, New York, 1980.

Rouse, W.B. Human-computer interaction in control of dynamic systems. ACM Computing Surveys, Vol. 13, No. 1, March, 1981, pp. 71-99.

Rouse, W. B. Models of human problem solving: detection, diagnosis, and compensation for system failures. Proceedings of IFAC Conference on Analysis, Design, Evaluation of Man-Machine Systems, Baden-Baden, F.R. of Germany, September 1982.

Rouse, W.B. and Morris, N.M. Human problem solving in a process control task. In Proc. of the First European Annual Conference on Human Decision Making and Manual Control, Delft, The Netherlands, 1981a, pp. 218-226.

Rouse, W.B. and Morris, N.M. Studies of human problem solving in a process control task. In Proc. 1981 IEEE Intl. Conf. on Cybern. and Society, Atlanta, 1981b, pp. 479-487.

Rouse, W.B., Rouse, S.H. and Pellegrino, S.J. A rule-based model of human problem solving performance in fault diagnosis tasks. IEEE Trans. Sys., Man, and Cybern. Vol. SMC-10, 1980, pp. 366-376.

Schank, R. C. and Abelson R. P. Scripts, plans, goals, and understanding. Hillsdale, New Jersey, Lawrence Erlbaum, 1977.

Sheridan. T.B. and Ferrell, W.B. Man-machine systems information, control, and decision making models of human performance, MIT-Press, Cambridge. MA, 1974.

Shortliffe, E.W. Computer-based medical consultations: MYCIN. Amsterdam: American Elsevier Publishing Co., Inc., 1976.

Waterman. D.A. and Hayes-Roth, F. (Eds.) Pattern-directed inference systems. New York, Academic Press, 1978.

Wesson, R.B. Planning in the world of the air traffic controller. Proceeding of the 5th International Joint Conference on Artificial Intelligence. Cambridge. MA, MIT Artificial Intelligence Laboratory, 1977.

Young, R.M. Production systems for modeling human cognition. In D.Michie (Ed) Expert systems in the microelectronic age. Edinburgh University Press, 1979, 35-43.

ND

8